第一周工作报告_曹金坤

作为小学期的第一周的总结，这篇报告介绍了第一周我参与的小组项目中自己所做的贡献。

调研数据获取接口和网站

因为我们的项目是建立一个学术论文的可视化和分享的网站，背后需要大量的相关数据支持，为了或许这些数据，我首先试图从一些机构的对外公开提供的api和sdk入手，毕竟这是最直观和方法的途径。我调研的网站包括但不限于：Google Scholar，Web of Science，dblp和IEEE Xplore等，对于从这几个网站可以获取的数据方法简介如下：

- Google Scholar：不支持以任何被认证的接口或者sdk获取其网站上的数据；
- Web of Science：支持在机构（比如交大）注册账号认证下下载特定论文的数据，但是数据下载量很小，且数据格式可读性差，只能在导入EndNote后获取其内部的信息；
- dblp：同Google Scholar，不提供公开的数据获取接口
- IEEE Xplore：对于开发者，提供免费的数据接口，但是对每日和每小时的api使用次数有限制。

学习和制作爬虫

综上所述，为了获取大量的学术论文相关信息，暂时没有找到合适的api。即使是IEEE Xplore等提供了对外接口的网站，其提供的数据量也很有限，而且数据的内容难以满足我们我的要求（具体来说，因为我们需要建立论文节点之间的引用关系图谱，所以需要不同论文之间的引用情况，但是已经得到的所有数据接口都只提供一篇文章的被引用次数，而不直接提供引用了这篇论文的论文目录）。所以，我转向利用自制爬虫的方法来获取所需要的论文数据。

综合了网络上的信息，我得知了对Google Scholar进行大量数据爬取是一个之前很多人尝试过但是没有太好结果的事情（归咎于Google的Bot检测系统），而且因为网络访问的原因，我放弃了对Google Scholar的爬取。之后衡量了不同网站的数据量和访问便捷性（比如Web of science的每次搜索访问都需要检查登录信息），最终我选择了IEEE Xplore作为我第一步数据爬虫的目标。

因为我之前对于爬虫技术也不是很熟悉，所以这次的工作开始后，在周一和周二两天，我主要进行了相关爬虫只是的学习，我用的教材是《Python3网络爬虫开发实战》。

第一开始，我企图使用urllib + requests + beautifulsoup的工具栈来对Xplore上的数据进行获取。但是发现了获取的HTML和预想的内容不同。之后查阅资料发现，这是因为Xplore上的大量数据都是在开启网页的时候通过js动态生成的。因此，我又转向了selenium + beautifulsoup + re的工具栈，通过模拟人的登录和操作动作来获取数据。

因为不同浏览器内核兼容性的缺陷，对于Xplore网站的点击会出现Bug，为了解决这种问题，爬虫分为了两个步骤，第一步是登录Xplore,输入关键词进行搜索，在网页加载了之后模拟用户点击、滑动和翻页，获取论文的展示页面信息。第二步是通过点击每篇论文的“Citation By Paper”按钮，进入引用了这篇论文的论文展示页面，在这个页面上再进行点击、滑动和翻页操作来得到引用论文的名录，之后查询这些论文，递归地进行论文数据的爬取。

在第一步中，我们可以获得论文的title，author，year，publisher，cited等数据，在第二部中可以得到引用了这篇论文的论文名录。第一步在webdriver.Chrome()内核上进行，第二部在PhantomJS上进行。

之后，因为要对抗IEEE的反爬虫策略和提高爬取的效率，我还在自己的机器上设置了定时的ip代理切换以及在爬虫程序中添加了多线程的操作方法。

截止7/6下午三点，目前已经获得了近万篇论文的前述数据。因为在进行对论文的关键词查询的时候，我选择的方法是将查询结果按照被引用次数进行降序的排列，所以在初期得到的论文都是被引用上千次乃至于上万次的论文，查询速度很慢，在之后，论文爬取的速度应该会加快。

数据接口的统一

因为需要对爬取的数据进行二次的处理，以满足数据库格式的需要（比如得到的数据中，author 和paper，paper 和paper都是多对多的关系，在mysql的默认单表中无法被存储，需要拆分），**我还编写了从原始的csv文件到满足数据库表格式需要的json文件的程序，使得新数据的获取、填充、上线和展示可以自动化的进行。**这一部分现在正在开发，即将完成。